Anexo H. Manual técnico.

MANUAL TÉCNICO

1. Instalación de eclipse:

- Descargar de la página https://www.eclipse.org/downloads/ el eclipse IDE for Java EE Developers.
- Poner el comprimido de eclipse en la carpeta raíz C://.
- Extraer la carpeta.
- Acceder a la carpeta de eclipse y dar clic en eclipse.exe.
- Guardar el proyecto en el workspace.

2. Importar proyecto:

- Clic en *File,*
- Clic en *Import.*
- Clic en *Existing Project into workspace*.
- Seleccionar ruta del Proyecto.
- Finalizar

3. Configurar ***Built Path:***

- Clic derecho en la carpeta del Proyecto.
- Clic en *built path*.
- Clic en configure *built path*.
- Seleccionar *Add JARs.*
- Selecciona todos los.jar de la carpeta JUNG.
- Clic en *Ok*.

4. Directorios:

- **Src (Source):** el directorio *src* contiene el código fuente de java (.java).
- **Img:** se encuentran todos los iconos que aparecen en la interfaz.
- **Data:** en este directorio se encuentra el archivo *graphml* que contiene el esquema de datos el cual se carga en la interfaz.
- **Record:** en este directorio se guarda temporalmente el esquema en la medida que el usuario realiza las operaciones de transformación del grafo.
- **Result:** guarda una consulta final en un archivo xml de respuesta.
- **Help:** se encuentra el manual del usuario almacenado en un archivo pdf.

- **Jung:** contiene la librería JUNG que permite diagramar y manipular el esquema de datos en la interfaz.

5. Paquetes y Clases:

El proyecto maneja dos paquetes que son *Graph* e *InterfaceG*. En el paquete de *Graph* se encuentran las clases *node* y *edge* los cuales definen la estructura de datos de los nodos y arcos del esquema de datos.

En el paquete *IntefaceG* se encuentran las clases relacionadas con la interfaz, específicamente la interacción, la lectura y la persistencia del esquema de datos.

Las clases de *InterfaceG* son:

- **writeGraphML.java**
  Esta clase permite escribir el esquema de datos en un archivo *graphml.*
- **record.java**
  Tiene como objetivo la persistencia de la secuencia de las operaciones que realice el usuario.
- **vertexLabel.java**
  Esta clase es un *implement* de la librería JUNG para personalizar las etiquetas de los nodos del grafo en la interfaz.
- **schemeInterface.java**
  Esta clase utiliza la librería JUNG para construir la interfaz de usuario, esta contiene el método *main*.
- **graphScheme.java**
  Lee el esquema desde un archivo *graphml* para cargarlo en la interfaz.
- **egdeFactory.java y vertexFactory.java**
  Se utilizan para la construcción de los nodos y arcos cuando la clase *graphScheme* instancia el esquema de datos desde el archivo *graphml.*
- **vertexLabelColor.java**
  Esta clase es un *implement* de la librería JUNG para personalizar el color de la etiquetas de los nodos.
- **picIcon.java**

Esta clase es un *implement* de la librería JUNG para asignar los iconos a los nodos y que dependen del tipo ya definido en el esquema de datos.

- **edgeLabel.java**
  Esta clase es un *implement* de la librería JUNG para personalizar las etiquetas de los arcos del grafo.
- **way.java**
  Esta clase encuentra los caminos que existen entre los nodos seleccionados.
- **result.java**
  Construye el archivo *xml* con la secuencia de los operadores utilizadas para hacer una consulta, es decir, genera el archivo de salida de la interfaz y la entrada al mediador.
- **step.java**
  Guarda el esquema de datos actual de cada secuencia, tales como los nodos, los arcos, las etiquetas, el operador utilizado y el nombre del archivo donde se guarda cada secuencia.

6. Archivo graphml:

A continuación se explica un ejemplo de archivo GraphML con un grafo esquema.

La etiqueta "*graphml*" representa el nodo raíz. El nodo "*key"* representa los atributos de los nodos (*node*) y arcos (*edge*) del grafo. Los atributos del nodo "*key*" son: el identificador "*id*" y "*for*" el cual muestra el tipo de atributo, en este caso es "*node*" o "*edge*".

```
<key id="Type" for="node"></key>
<key id="Label" for="node"></key>
<key id="Edges" for="edge"></key>
```

El nodo "*graph*" define los nodos y los arcos del grafo, este tiene como atributos el identificador "*id*" y "*edgedefault*" el cual indica el tipo de grafo, en este caso su valor seria *undirected*.

```
<graph id="G" edgedefault="undirected">
```

Un nodo del grafo se define con el nombre "*node*", el cual tiene como atributo el identificador "*id*". Dentro de las etiquetas "*node*" se encuentran dos nodos "data" el cual tiene un atributo "*key*" que hace referencia a los atributos definidos para

“*node*”. La definición del nodo en el archivo está dada de acuerdo a la estructura del nodo en el grafo esquema.

Los atributos son *Type, Value, Operation Label*. El atributo *Type* define el tipo de nodo: objeto, compuesto, básico, etc. El atributo *Value* es el valor que ingresa un usuario como parámetro de condición, en el esquema de entrada siempre es vacío hasta que se ingrese el valor de condición. El atributo *Operation* es el valor de la operación que ingresa un usuario como parámetro de condición, esta puede ser (=, <, >, <=, >=, <>), también en el esquema de entrada siempre se encuentra vacío hasta que se ingrese su valor. El atributo *Label* representa la etiqueta del nodo. Los atributos *Value* y *Operation* se adicionaron en el archivo GraphML dadas las características de los nodos en el grafo esquema.

```
<node id=’1’>
        <data key=’Type’>objeto</data>
        <data key=’Value’></data>
        <data key=’Operation’></data>
        <data key=’Label’>Nodo 1</data>
</node>
```

Los arcos se definen con el nombre “*edge*”. Este tiene como atributos el identificador “*id*”, “*source*” que muestra un nodo inicio y “target” que indica el nodo destino del arco. Tiene un nodo “*data*” con un atributo “*key*” el cual hace referencia a los atributos definidos para “*edge*”.

```
<edge id=’4’ source=’1’ target= ’2’ >

<data key=’Edges’>Arco 1</data>

</edge>
```

Un ejemplo de este archivo GraphML se observa a continuación:

```
<?xml version="1.0" encoding="UTF-8"?>

<graphml xmlns=http://graphml.graphdrawing.org/xmlns
xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:schemaLocation="http://graphml.graphdrawing.org/xmlns
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd">

        <key id="Type" for="node"></key>
        <key id="Label" for="node"></key>
        <key id="Edges" for="edge"></key>
        <key id="Operation" for="node"><default></default></key>
```

```
        <key id="Edges" for="edge"><default></default></key>

    <graph id="G" edgedefault="undirected">
        <node id='1'>
            <data key='Type'>objeto</data>
            <data key='Value'></data>
            <data key='Operation'></data>
            <data key='Label'>Nodo 1</data>
        </node>
        <node id='2'>
            <data key='Type'>compuesto</data>
            <data key='Value'></data>
            <data key='Operation'></data>
            <data key='Label'>Nodo 2</data>
        </node>
        <node id='3'>
            <data key='Type'>basico</data>
            <data key='Value'></data>
            <data key='Operation'></data>
            <data key='Label'>Nodo 3</data>
        </node>
        <edge id='4' source='1' target= '2' >
            <data key='Edges'>Arco 1</data>
        </edge>
        <edge id='5' source='2' target= '3' >
            <data key='Edges'>Arco 2</data>
        </edge>

    </graph>

</graphml>
```

7. **Archivos xml:**

Cuando se genera la consulta final se obtiene un archivo XML con los arcos que forman un camino entre los nodos y la secuencia de los operadores.
El archivo XML para la operación extraer contiene la secuencia <secuencia> con un identificador de secuencia <idSec> el cual indica el orden de la secuencia de

los operadores. El siguiente es el operador <operador> que indica qué operador utilizó, por último se pone el camino a través de los arcos <arco>. A continuación se observa un ejemplo de extraer subgrafo.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<consulta>
  <secuencia>
        <idSec>1</idSec>
        <operador>EXTRACT</operador>
        <camino>
              <arco><id>A1</id></arco>
              <arco><id>A2</id></arco>
              <arco><id>A3</id></arco>
        </camino>
  </secuencia>
 </consulta>
```

El archivo XML para el operador contraer camino contiene las mismas especificaciones de extraer subgrafo, a excepción que contiene en la etiqueta <nodoNuevo > los nuevos arcos y el nuevo nodo.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<consulta>
        <secuencia>
              <idSec>1</idSec>
              <operador>CONTRACTION</operador>
              <camino>
                    <arco><id>A1</id></arco>
                    <arco><id>A2</id></arco>
                    <arco><id>A3</id></arco>
                    <arco><id>A4</id></arco>
                    <arco><id>A5</id></arco>
              </camino>
              <nodoNuevo>
                    <arco><id>NuevoA1</id></arco>
                    <arco><id>NuevoA2</id></arco>
                    <node><id>NuevoNODO</id></node>
              </nodoNuevo>
        </secuencia>
```

</consulta>

El archivo XML para el operador filtro de valor contiene las mismas especificaciones que extraer subgrafo, a excepción que entre las etiquetas <camino> y </camino> se encuentran los caminos entre un nodo y el nodo que tiene el valor de condición. <node> indica el nodo donde inicia un camino y entre las etiquetas <pk> y </pk> se encuentran los arcos que forman dicho camino. Al final se encuentra la etiqueta <filtro> en el cual se indican la condición o las condiciones <condicion>. Entre las etiquetas <condicion> y </condicion> se encuentra la información del nombre del nodo <node>, el operador <operador> y el valor de condición <valor> dada a dicho. A continuación se muestra un ejemplo del XML para filtro de valor.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<consulta>
	<secuencia>
		<idSec>1</idSec>
		<operador>VALUEFILTER</operador>
			<camino>
			<node>Nodo1</node>
				<pk>
					<arco><id>A1</id></arco>
					<arco><id>A2</id></arco>
					<arco><id>A3</id></arco>
				</pk>
			<node>Nodo2</node>
				<pk>
					<arco><id>A4</id></arco>
					<arco><id>A5</id></arco>
					<arco><id>A6</id></arco>
					<arco><id>A7</id></arco>
				</pk>
			</camino>
			<filtro>
				<condicion>
					<node>Nodo3</node>
					<operador>=</operador>
					<valor>Valor de condición para A1</valor>
```

```
                    </condicion>
                    <condicion>
                        <node>Nodo4</node>
                        <operador>=</operador>
                        <valor>Valor de condición para A4</valor>
                    </condicion>
                </filtro>
    </secuencia>
</consulta>
```

El archivo XML para el operador filtro de clase contiene en <camino> el nodo objeto de interés en <node>. Entre las etiquetas <pk> se encuentra los arcos que forman un camino entre el nodo objeto y el nodo con el valor de condición. Al final se encuentra entre el filtro <filtro> y condición <condicion> los nodos <node>, los operadores <operador> y los valores de condición <valor>. A continuación se observa un ejemplo de filtro de clase.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>

<consulta>
    <secuencia>
        <operador> CLASSFILTER</operador>
        <idSec>1</idSec>
        <camino>
            <node><id>NodoObjeto1</id></node>
                <pk>
                    <arco><id>A1</id></arco>
                    <arco><id>A2</id></arco>
                    <arco><id>A3</id></arco>
                </pk>
        </camino>
        <filtro>
            <condicion>
                <node>Nodo1</node>
                <operador>=</operador>
                <valor>Valor de condición para A1</valor>
            </condicion>
        </filtro>
    </secuencia>
```

</consulta>

A continuación se observa un ejemplo con una secuencia de operadores. Se realiza primero extraer subgrafo con el id de secuencia 1 y luego filtro de valor con el id de secuencia en 2.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<consulta>
    <secuencia>
        <idSec>2</idSec>
        <operador>VALUEFILTER</operador>
        <camino>
            <node>Nodo1</node>
            <pk>
                <arco><id>A1</id></arco>
                <arco><id>A2</id></arco>
                <arco><id>A3</id></arco>
                <arco><id>A4</id></arco>
            </pk>
            <node>Nodo2</node>
            <pk>
                <arco><id>A5</id></arco>
                <arco><id>A6</id></arco>
                <arco><id>A7</id></arco>
            </pk>
        </camino>
        <filtro>
            <condicion>
                <node>Nodo3</node>
                <operador>=</operador>
                <valor>ValordeCondición1</valor>
            </condicion>
            <condicion>
                <node>Nodo4</node>
                <operador>=</operador>
                <valor>123456789</valor>
            </condicion>
        </filtro>
    </secuencia>
```

```
    <secuencia>
        <idSec>1</idSec>
                <operador>EXTRACT</operador>
                        <camino>
                                <arco><id>A1</id></arco>
                                <arco><id>A2</id></arco>
                                <arco><id>A3</id></arco>
                                <arco><id>A4</id></arco>
                                <arco><id>A5</id></arco>
                                <arco><id>A6</id></arco>
                                <arco><id>A7</id></arco>
                        </camino>
    </secuencia>
</consulta>
```